

ปฏิบัติการที่ 1: ติดตั้งเครื่องมือและสร้างหน้าเว็บแรก

เป้าหมาย: ติดตั้งเครื่องมือพัฒนาและสร้างหน้าเว็บแรกพร้อมการใช้ Git

ผลลัพธ์การเรียนรู้:

- ติดตั้งสภาพแวดล้อมพัฒนาแบบสมบูรณ (VS Code, Git, Node.js)
- สร้างโปรเจกต์ HTML/CSS/JavaScript แรก
- ใช้ Browser DevTools เพื่อการแก้ไขปัญหา
- สามารถใช้ Git workflow พร้อมข้อความ commit ที่มีความหมาย
- อัปโหลดโค้ดไป GitHub

ข้อกำหนดเบื้องต้น

- คอมพิวเตอร์ที่ใช้ Windows/Mac/Linux
- การเชื่อมต่ออินเทอร์เน็ต
- บัญชี GitHub (หรือสร้างในระหว่างปฏิบัติการ)

ปฏิบัติการ (Lab) - 5 กิจกรรม พร้อม Git

กิจกรรมที่ 1: ติดตั้งเครื่องมือ

วัตถุประสงค์: ติดตั้ง VS Code, Node.js, และ Git

ขั้นตอน

1. ดาวน์โหลดและติดตั้ง

- [Visual Studio Code](#)
- [Node.js LTS](#)
- [Git](#)

2. ตรวจสอบการติดตั้ง (PowerShell/Command Prompt) โดยเปิด Terminal

```
node --version
npm --version
git --version
code --version
```

ผลที่คาดหวัง (เวอร์ชันเหล่านี้หรือใหม่กว่า)

```
v18.17.0 (หรือเวอร์ชันใหม่กว่า)
```

```
9.6.7
```

```
git version 2.40.0...
```

```
2.20.0...
```

3. ตั้งค่า Git (ครั้งแรกเท่านั้น)

```
git config --global user.name "ชื่อของคุณ"
```

```
git config --global user.email "email@example.com"
```

```
git config --global --list # ตรวจสอบ
```

```
C:\Users\witta>git config --global --list
core.editor="C:\Users\witta\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.email=wittawas@informatics.buu.ac.th
user.name=wittawas@informatics.buu.ac.th
```

รายการตรวจสอบ

- ติดตั้ง VS Code, Node.js, และ Git สำเร็จ
- ทดสอบเวอร์ชันทั้งหมด
- ตั้งค่า Git ด้วยชื่อและอีเมล

กิจกรรมที่ 2: สร้างโปรเจกต์และเริ่ม Git

วัตถุประสงค์: สร้างโฟลเดอร์โปรเจกต์และเริ่มต้น Git repository

ขั้นตอน

1. สร้างโฟลเดอร์และเปิด Terminal

สร้างโฟลเดอร์

```
mkdir wk01-lab
cd wk01-lab
```

เปิด VS Code

```
code .
```

2. เริ่มต้น Git Repository: เปิด New Terminal และพิมพ์คำสั่งต่อไปนี้

```
git init
git config user.name "ชื่อของคุณ"
git config user.email "email@example.com"
```

3. สร้างไฟล์ .gitignore (VS Code)

```
node_modules/
.DS_Store
*.log
```

4. Commit ครั้งแรก (บันทึกการเปลี่ยนแปลง):

```
git add .gitignore
git commit -m "Initial commit: Add gitignore"
```

```
git log --oneline # ตรวจสอบ
```

รายการตรวจสอบ

- สร้างโฟลเดอร์ wk01-lab สำเร็จ
 - รัน git init เสร็จแล้ว
 - ตั้งค่า Git user (ระดับ global หรือ local)
 - บันทึก (Commit) ไฟล์ .gitignore
-

กิจกรรมที่ 3: สร้างหน้า HTML แรก

วัตถุประสงค์: สร้างหน้า HTML พื้นฐานพร้อมการใช้ Git

ขั้นตอน:

1. สร้างไฟล์ index.html (ใน VS Code)

```
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>เว็บแรกของฉัน</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      min-height: 100vh;
      display: flex;
      align-items: center;
      justify-content: center;
      margin: 0;
    }

    .container {
      background: white;
      padding: 2rem;
      border-radius: 12px;
      box-shadow: 0 10px 30px rgba(0, 0, 0, 0.3);
      text-align: center;
      max-width: 500px;
    }

    h1 {
      color: #667eea;
      margin-bottom: 1rem;
    }

    p {
      color: #666;
      line-height: 1.6;
      margin-bottom: 1.5rem;
    }

    img {
      max-width: 150px;
      border-radius: 50%;
      margin: 1rem 0;
      border: 4px solid #667eea;
    }

    button {
      background: #667eea;
      color: white;
      padding: 0.75rem 1.5rem;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      font-size: 1rem;
      transition: background 0.3s;
    }

    button:hover {
      background: #764ba2;
    }
  </style>

```

```

    #message {
      margin-top: 1rem;
      padding: 1rem;
      background: #f0f0f0;
      border-radius: 4px;
      display: none;
      color: #333;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>สวัสดี! ยินดีต้อนรับ</h1>
    <p>นี่คือหน้าเว็บแรกของฉันที่สร้างขึ้นด้วย HTML5 & CSS</p>

    <button id="btn">คลิกฉัน!</button>
    <div id="message">ยินดีต้อนรับสู่โลกของ Web Development</div>
  </div>

  <script>
    document.getElementById("btn").addEventListener("click", function () {
      const msg = document.getElementById("message");
      msg.style.display = msg.style.display === "none" ? "block" : "none";
    });
  </script>
</body>
</html>

```

2. บันทึกไฟล์ โดยกด Ctrl+S

3. Commit ครั้งที่สอง (บันทึกลงไป Git)

```
git add index.html
```

```
git commit -m "Add: First HTML page with styling and JavaScript
```

```
- HTML5 structure with meta tags
```

```
- CSS styling for responsive design
```

```
- JavaScript button interaction
```

```
- Gradient background and card layout"
```

รายการตรวจสอบ

- สร้างไฟล์ index.html ครบถ้วน
- ทดสอบเปิดดูในเบราว์เซอร์
- บันทึก (Commit) ไป Git

กิจกรรมที่ 4: DevTools และการปรับปรุง

วัตถุประสงค์: ปรับปรุงการเขียนโค้ดโดยแยก CSS และ JavaScript ออกจาก HTML และใช้ DevTools

ขั้นตอน

1. สร้างไฟล์ styles.css (แยก CSS ออกจาก HTML)

```
* {
  margin: 0;
  padding: 0;
}
```

```
    box-sizing: border-box;
}

body {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
}

.container {
  background: white;
  padding: 2rem;
  border-radius: 12px;
  box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);
  text-align: center;
  max-width: 600px;
  width: 90%;
}

h1 {
  color: #667eea;
  margin-bottom: 1rem;
  font-size: 2rem;
}

.profile-info {
  margin: 2rem 0;
}

.profile-info p {
  color: #666;
  line-height: 1.8;
  margin-bottom: 1rem;
}

img {
  max-width: 120px;
  border-radius: 50%;
  border: 4px solid #667eea;
  margin: 1rem 0;
}

.button-group {
  display: flex;
  gap: 1rem;
  justify-content: center;
  margin: 1.5rem 0;
}

button {
  background: #667eea;
  color: white;
  padding: 0.75rem 1.5rem;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 1rem;
  transition: all 0.3s;
}

button:hover {
  background: #764ba2;
  transform: translateY(-2px);
}
```

```

    box-shadow: 0 4px 12px rgba(102, 126, 234, 0.4);
}

#message {
    margin-top: 1rem;
    padding: 1rem;
    background: #e8f0ff;
    border-left: 4px solid #667eea;
    border-radius: 4px;
    display: none;
    color: #333;
    animation: slideIn 0.3s ease-out;
}

@keyframes slideIn {
    from {
        opacity: 0;
        transform: translateY(-10px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

@media (max-width: 600px) {
    .container {
        padding: 1.5rem;
    }

    h1 {
        font-size: 1.5rem;
    }

    .button-group {
        flex-direction: column;
    }
}

```

2. สร้างไฟล์ script.js:

```

// Show/hide message
document.getElementById("btn-show").addEventListener("click", function () {
    const msg = document.getElementById("message");
    msg.style.display = msg.style.display === "none" ? "block" : "none";
});

```

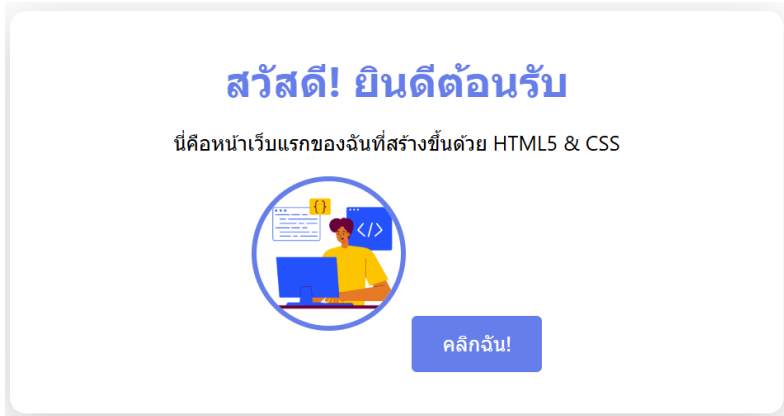
3. อัปเดต index.html (เอา <style> ออก เพิ่ม <link>, แก้ไขส่วน <script>)

```

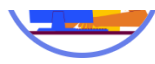
<!DOCTYPE html>
<html lang="th">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>เว็บแรกของฉัน</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <div class="container">
      <h1>สวัสดี! ยินดีต้อนรับ</h1>
      <p>นี่คือน้ำเว็บแรกของฉันที่สร้างขึ้นด้วย HTML5 & CSS</p>

      
    <button id="btn">คลิกฉัน!</button>
    <div id="message">ยินดีต้อนรับสู่โลกของ Web Development</div>
</div>

<script src="script.js"></script>
</body>
</html>
```



คลิกปุ่ม คลิกฉัน! จะปรากฏข้อความ ยินดีต้อนรับสู่โลกของ Web Development



คลิกฉัน!

ยินดีต้อนรับสู่โลกของ Web Development

4. เปิด **DevTools** โดยกด F12 หรือ Right-click > Inspect

1. **Elements tab:** ดู HTML structure
2. **Network tab:** ดู file loading
3. **Console tab:** ทดสอบ JavaScript

5. **Commit** ครั้งที่สาม:

```
git add styles.css script.js index.html
git commit -m "Refactor: Separate HTML, CSS, and JavaScript
- Extract styles.css from index.html
- Create script.js for JavaScript logic
- Add media queries for responsive design
- Improve button functionality with reset button
- Add CSS animations"
```

รายการตรวจสอบ

- สร้างไฟล์ styles.css
- สร้างไฟล์ script.js
- อัปเดต index.html ให้เชื่อมโยงไฟล์
- ทดสอบใน DevTools
- บันทึก (Commit) ไป Git

กิจกรรมที่ 5: สรุปและ Commit สุดท้าย

วัตถุประสงค์: รวบรวมและสรุปบันทึก

ขั้นตอน:

1. สร้าง README.md และพิมพ์เนื้อหาต่อไปนี้

สัปดาห์ที่ 1 - Tools & First Web Page

ไฟล์ที่สร้าง

- **index.html** - HTML structure with semantic elements
- **styles.css** - CSS styling with responsive design
- **script.js** - JavaScript for interactivity
- **README.md** - Project documentation

เรียนรู้

- Tools setup (VS Code, Node.js, Git)
- HTML5 structure
- CSS3 styling and media queries
- JavaScript DOM manipulation
- Git workflow and commits

Git Commits

1. Initial commit: .gitignore
2. First HTML page with styling
3. Refactor: Separate CSS and JavaScript
4. Summary: Week 1 complete

How to Run

1. เปิด index.html in browser
2. หรือเปิดด้วย Live Server in VS Code

Next Steps

- Week 2: HTML5 Semantic & Forms
- Week 3: CSS Flexbox Layouts

2. Commit ครั้งสุดท้าย

- ```
git add README.md
git commit -m "Add: Week 1 summary and documentation"
```
- Completed tool installation
  - Created first HTML page
  - Separated HTML, CSS, JavaScript
  - Learned Git workflow with meaningful commits
  - Ready for Week 2"

### 3. ดู Git history (ประวัติการเปลี่ยนแปลง)

```
git log --oneline --all
```

ผลที่คาดหวัง

```
4a8c9d2 (HEAD -> main) Add: Week 1 summary
3f2b1e5 Refactor: Separate CSS and JavaScript
c7e4a9f Add: First HTML page with styling
8a3d2c1 Initial commit: Add gitignore
```

#### 4. ตรวจสอบไฟล์:

git status # ควร clean

ls -la # ดู files

#### รายการตรวจสอบ

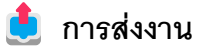
- สร้าง README.md
  - Commit สุดท้าย
  - ตรวจสอบ commit history
  - ทั้ง 4 ไฟล์อยู่ใน Git
- 

#### สรุปสิ่งที่ได้เรียนรู้ในปฏิบัติการ

1. การติดตั้งเครื่องมือ
  - a. VS Code - code editor
  - b. Node.js - runtime environment
  - c. Git - version control
2. Git Workflow
  - a. git init - initialize repository
  - b. git add - stage files
  - c. git commit - save changes
  - d. git log - view history
  - e. Meaningful commit messages
3. First Web Page
  - a. HTML5 structure
  - b. CSS styling & responsive
  - c. JavaScript interactivity
  - d. การใช้ DevTools
4. File Organization
  - a. Separate HTML, CSS, JS
  - b. .gitignore for unnecessary files
  - c. README.md สำหรับ documentation

#### คำถามสำหรับคิด

1. ทำไมต้องใช้ Git ในการพัฒนาเว็บ?
  2. ความสำคัญของการเขียน commit message ที่ชัดเจน?
  3. DevTools ช่วยอะไรในการพัฒนาเว็บ?
  4. ทำไมถึงต้องแยก HTML, CSS, JavaScript?
-



## การส่งงาน

### 1. ส่งไปยัง GitHub Repository

1. สร้าง repository ชื่อ wk01-lab ที่ GitHub.com
2. Description: คำอธิบายโปรเจกต์นี้
3. เลือก Public
4. Create repository

### 2. เชื่อมต่อ remote

git remote add origin https://github.com/usernameของเรา/wk01-lab.git

### 3. Push local repo ไปยัง Github (remote)

git push -u origin master

### 4. ที่ Github /wk01-lab จะปรากฏไฟล์

The screenshot shows a GitHub repository page for 'wk01-lab' (Public). The repository is owned by 'wittawasatinformaticsbuuacth'. The files listed are:

| File Name  | Commit Message                               | Time           |
|------------|----------------------------------------------|----------------|
| .gitignore | initial commit: add gitignore                | 1 hour ago     |
| README.md  | Add: Week 1 summary and documentation        | 1 hour ago     |
| index.html | Refactor: Separate HTML, CSS, and JavaScript | 1 hour ago     |
| script.js  | Refactor: Separate HTML, CSS, and JavaScript | 1 hour ago     |
| styles.css | short                                        | 22 minutes ago |

The README file is expanded, showing the following content:

### สัปดาห์ที่ 1 - Tools & First Web Page

#### ไฟล์ที่สร้าง

- index.html - HTML structure with semantic elements
- styles.css - CSS styling with responsive design
- script.js - JavaScript for interactivity
- README.md - Project documentation

#### เรียนรู้

- Tools setup (VS Code, Node.js, Git)
- HTML5 structure

5. ส่งไฟล์ รหัสนิสิต-lab01.docx ที่มี repository url https://github.com/usernameของเรา/wk01-lab.git พร้อม capture screen Github ตามตัวอย่างในข้อ 4 ใน Google Classroom

## ก่อนส่งงาน ให้ตรวจสอบดังนี้

### ไฟล์ที่สร้าง

- index.html (ที่มี HTML5 structure)
- styles.css (ที่มี styling)
- script.js (ที่มี JavaScript)
- README.md (ที่มี documentation)
- ไฟล์ .gitignore (ในโฟลเดอร์)

### คุณภาพโค้ด

- ไม่มี HTML errors (F12 → Console ไม่มี errors)
- CSS เชื่อมโยงอย่างถูกต้อง
- JavaScript เชื่อมโยงอย่างถูกต้อง

### Git Workflow

- 4+ commits ที่มีข้อความที่มีความหมาย
- Commit history แสดงความก้าวหน้า
- ไฟล์ทั้งหมด staged และ committed

### เอกสาร

- README.md อธิบายโปรเจกต์
- Comments ในโค้ดตามที่จำเป็น
- Commit messages ชัดเจนและอธิบายได้

### การทดสอบ

- หน้าเปิดในเบราว์เซอร์โดยไม่มี errors
- ปุ่ม/interaction ทำงาน
- DevTools ไม่มี errors

ปัญหา: **"git command not found"**

วิธีแก้ไข:

1. Reinstall Git from git-scm.com
2. Make sure to add to PATH
3. Restart Terminal/PowerShell
4. Check: git --version

ปัญหา: **"Permission denied (publickey)"**

วิธีแก้ไข:

1. This happens if using SSH instead of HTTPS
2. Either:
  - a) Use HTTPS URL instead of SSH
  - b) OR setup SSH keys (advanced)
3. For now, use HTTPS

ปัญหา: **"fatal: not a git repository"**

วิธีแก้ไข:

1. Make sure you're in the right folder
2. Run: pwd (Mac/Linux) or cd (Windows)
3. Run: git init (to create .git folder)
4. Then: git add .

ปัญหาเบราว์เซอร์

ปัญหา: หน้าแสดง **404** หรือว่าง

วิธีแก้ไข:

1. Make sure index.html exists
2. Check file path is correct
3. Use Live Server extension
4. Try: http://127.0.0.1:5500/index.html

ปัญหา: สไตลไม่แสดง

วิธีแก้ไข:

1. Check: <link rel="stylesheet" href="styles.css">
2. Check spelling of "styles.css"
3. Check both files in same folder
4. Hard refresh: Ctrl+Shift+R (or Cmd+Shift+R Mac)
5. Open DevTools → Network tab → check CSS status

ปัญหา: **JavaScript** ไม่ทำงาน

วิธีแก้ไข:

1. Open DevTools → Console tab
2. Check for error messages
3. Verify: <script src="script.js"></script> at end of body
4. Check spelling of "script.js"
5. Use console.log() to debug

ข้อผิดพลาดทั่วไปและวิธีแก้ไข

| Error                                            | Cause                                 | Solution                                    |
|--------------------------------------------------|---------------------------------------|---------------------------------------------|
| Uncaught ReferenceError: variable is not defined | Missing variable                      | Check spelling, declare with let/const/var  |
| 404 Not Found                                    | File doesn't exist                    | Check file name, path, spelling             |
| Unexpected token <                               | HTML in JS file                       | Make sure script.js is JavaScript, not HTML |
| CORS error                                       | Trying to fetch from different domain | Use CORS proxy or local server              |
| git: command not found                           | Git not installed or not in PATH      | Reinstall Git, add to PATH                  |

## คำถามที่พบบ่อย (FAQ)

### Q: ต้องใช้ GitHub บังคับไหม?

A: ไม่บังคับ แต่แนะนำมากสำหรับ:

- Version control
- Backup
- Portfolio
- Collaboration (ในอนาคต)

สำหรับปฏิบัติการนี้ Git ท้องถิ่นเป็นสิ่งที่ต้องมี

### Q: ใช้ IDE ออนไลน์เช่น CodePen แทนได้ไหม?

A: ไม่แนะนำ เพราะ:

- ไม่สอน Git workflow
- ไม่สอน local development
- ต่างจาก professional workflow

ใช้ VS Code ท้องถิ่น + Git

### Q: ต้องมี commit กี่อัน?

A: ขั้นต่ำ 4:

1. Initial (.gitignore)
2. Add HTML
3. Add CSS/JS
4. Summary/docs

ถ้า Commit มากกว่านี้ = ดีกว่า (แสดงถึงมีการพยายามทำด้วยตนเอง)

### Q: สามารถคัดลอกโค้ดจากบทช่วยสอนออนไลน์ได้ไหม?

A: ได้:

- อ้างอิงและเรียนรู้จากบทช่วยสอน
- ใช้ code snippets เพื่อการเรียนรู้
- แก้ไขและปรับปรุงโค้ด

ไม่ได้:

- คัดลอกตรงๆ โดยไม่เข้าใจ
- ไม่ระบุแหล่งที่มา
- ส่งโดยไม่มีการแก้ไข

ดีที่สุด: เข้าใจแนวคิด → เขียนโค้ดเอง → ระบุแหล่งที่มา

### Q: ต้องอัปเดตหน้าเว็บไปออนไลน์บังคับไหม?

A: ไม่ สำหรับ

- เรียนรู้
- การทดสอบ
- การพัฒนา

ตั้งแต่ Week 5 เราจะ deploy ไป GitHub Pages

### Q: ถ้าทำผิดใน commit?

A: มีหลายตัวเลือก

1. git commit --amend (แก้ commit ล่าสุด)
2. git revert (ยกเลิก commit เฉพาะ)
3. git reset (ย้อนกลับ)

ตอนนี้สร้าง commit ใหม่พร้อมการแก้ไข

```
git commit -m "Fix: แก้บั๊ก"
```

### Q: โครงสร้างไฟล์เดอร์ของฉันต่างออกไปได้ไหม?

A: ได้ ตราบใดที่มี

- index.html
- styles.css

- script.js
- README.md
- โฟลเดอร์ .git ที่มี history

### 💡 เคล็ดลับ Git

```
See all commits with pretty view
git log --oneline --graph --all

Undo last change (before staging)
git restore filename.txt

View what changed since last commit
git diff

Stage specific file
git add specific-file.html

Commit with longer description
git commit -m "Title" -m "Detailed description here"
```

### 💡 เคล็ดลับ DevTools

- คำสั่ง Console ที่มีประโยชน์:
 

```
document.title // ได้ชื่อหน้า
document.body.innerHTML // ได้ HTML ทั้งหมด
localStorage.getItem('key') // ได้ค่าที่จัดเก็บ
fetch('url').then() // ทดสอบ API
```
- การกระทำของ Elements ที่มีประโยชน์:
 

```
Right-click element → "Break on" → ตั้ง breakpoint
Edit styles live เพื่อทดสอบ CSS
Copy HTML สำหรับอ้างอิง
```
- ตัวกรอง Network ที่มีประโยชน์:
 

```
Click "XHR" เพื่อดู API calls
Filter by status code (200, 404, ฯลฯ)
Check response tab สำหรับ JSON data
```

### เคล็ดลับ HTML/CSS

```
<!-- ต้องรวม meta tags เหล่านี้เสมอ -->
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<meta name="description" content="คำอธิบายหน้า" />

<!-- ใช้ semantic tags สำหรับโครงสร้างที่ดีขึ้น -->
<header>
 <!-- ส่วนหัวหน้า -->
 <nav>
 <!-- การนำทาง -->
 </nav>
 <main>
 <!-- เนื้อหาหลัก -->
 <article>
 <!-- บทความ/โพสต์ -->
 <aside>
 <!-- ไซด์บาร์ -->
 </aside>
 <footer><!-- ส่วนท้ายหน้า --></footer>
 </article>
 </main>
</header>
```

## แหล่งข้อมูล

### เอกสารอย่างเป็นทางการ

- [เอกสาร VS Code](#)
- [เอกสาร Git](#)
- [แนวทาง GitHub](#)
- [MDN: HTML](#)
- [MDN: CSS](#)
- [MDN: JavaScript](#)

### แหล่งข้อมูลการเรียนรู้

- [freeCodeCamp - Responsive Web Design](#)
- [web.dev - เรียนรู้การออกแบบเว็บ](#)
- [Codecademy - Git & GitHub](#)
- [Khan Academy - อินเทอร์เน็ต 101](#)

### บทช่วยสอน

- [บทช่วยสอน Git โดย Atlassian](#)
- [GitHub Skills - การฝึกอบรมฟรี](#)
- [YouTube: Web Dev Setup \(ค้นหา\)](#)